

Claims:

What is claimed is:

1. A system for code generation from a software application design
5 product source data, comprising:
a data navigation layer for interface with, and for providing navigational
access to, a software application design product source data;
a template for specifying a code generation process to be applied to
said source data;
10 a parser for parsing said template in accordance with any specified
rules, filter, conditions, and notifiers, to generate code; and,
a code output mechanism for the output said generated code.
2. The system of claim 1 wherein said code output mechanism outputs
15 said generated code to a storage device.
3. The system of claim 1 further comprising:
rules that implement template instructions and dynamically generate
output when static template code is not appropriate.
20
4. The system of claim 3 further comprising:
notifiers that include logic applied when a rule is invoked, to allow
external components to be notified of the progress of the code generation
process.
25
5. The system of claim 3 further comprising:
condition specifiers that include logic applied when a rule is invoked, to

evaluate conditions and allow code generation depending on specific conditions.

6. The system of claim 3 further comprising:
5 filters that include logic applied when a rules is invoked, to transform data.
7. The system of claim 1 wherein said system further includes:
internal rules that provide basic functions to query symbol values from
10 the data source, navigate through the data source, and open and close files.
8. The system of claim 1 wherein said system further includes:
internal filters that provide generic transformation capabilities, such as
lowercase/upercase conversion.
15
9. The system of claim 1 wherein said navigation layer allows mapping of
an abstracted data representation to said source data.
10. The system of claim 9 wherein said parser provides functions to
20 manipulate a scope stack, wherein said scope stack addresses said abstracted data representation.
11. The system of claim 10 wherein said parser creates a hierarchical
scope stack.
25
12. The system of claim 10 wherein navigation within said scope stack is
by a pointer.

13. A method of generating computer code, comprising the steps of:
providing a data navigation layer for interface with, and for providing
navigational access to, a software application design product source data;
providing a template for specifying a code generation process to be
5 applied to said source data;
parsing said template in accordance with any specified rules, filter,
conditions, and notifiers, to generate code; and,
outputting, via a code output mechanism, said generated code.
- 10 14. The method of claim 13 wherein said step of outputting includes
outputting said generated code to a storage device.
- 15 15. The method of claim 13 further comprising:
parsing rules that implement template instructions and dynamically
generate output when static template code is not appropriate.
- 20 16. The method of claim 15 further comprising:
parsing notifiers that include logic applied when a rule is invoked, to
allow external components to be notified of the progress of the code
generation process.
- 25 17. The method of claim 15 further comprising:
parsing condition specifiers that include logic applied when a rule is
invoked, to evaluate conditions and allow code generation depending on
specific conditions.

18. The method of claim 15 further comprising:
parsing filters that include logic applied when a rules is invoked, to
transform data.
- 5 19. The method of claim 13 further comprising the step of:
parsing internal rules that provide basic functions to query symbol
values from the data source, navigate through the data source, and open and
close files.
- 10 20. The method of claim 13 further comprising the step of:
parsing internal filters that provide generic transformation capabilities,
such as lowercase/uppercase conversion.
- 15 21. The method of claim 13 wherein said navigation layer maps an
abstracted data representation to said source data.
- 20 22. The method of claim 21 wherein said parser provides functions to
manipulate a scope stack, wherein said scope stack addresses said
abstracted data representation.
- 25 23. The method of claim 22 wherein said parser creates a hierarchical
scope stack.
24. The method of claim 22 further comprising the step of:
navigating within said scope stack using a pointer.
- 25 25. A system for code generation, comprising:

a data navigation layer for interface with, and for providing navigational access to, a software application design product source data, said navigation layer allows mapping of an abstracted data representation to said source data;

5 a template for specifying a code generation process to be applied to said source data;

a parser for parsing said template in accordance with any specified rules, filter, conditions, and notifiers, to generate code, said parser provides functions to manipulate a scope stack, wherein said scope stack addresses
10 said abstracted data representation, said parser creates a hierarchical scope stack, navigation within said scope stack is by a pointer;

rules that implement template instructions and dynamically generate output when static template code is not appropriate;

notifiers that include logic applied when a rule is invoked, to allow
15 external components to be notified of the progress of the code generation process;

condition specifiers that include logic applied when a rule is invoked, to evaluate conditions and allow code generation depending on specific conditions; and,

20 filters that include logic applied when a rules is invoked, to transform data.